A Confidence-Based Approach for Balancing Fairness and Accuracy

submitted for blind review

***Abstract*—We study three classical algorithms in the context of fairness: adaptive boosting, support vector machines, and logistic regression. Our goal is to maintain the high accuracy of these learning algorithms while reducing the degree to which they discriminate against individuals because of their membership in a certain protected group. A common feature of these learning algorithms is that one can easily measure their confidence in the classification of a point. Our main contribution is a method for achieving fairness by shifting the confidence threshold for the protected group. We compare this method with other "fair" variants of these learning algorithms as well as results in previous papers in the fairness literature. Our method, in addition to outperforming the state of the art in terms of achieving high accuracy and low discrimination, also allows for a fast and simple quantification of the trade-off between bias and error. We study this trade-off for all three methods on three different datasets. We further define a new notion of fairness by introducing a modeling assumption on the process generating bias in the training data, and we evaluate our methods according to this measure.**

Machine learning algorithms assume an increasingly large role in making decisions across business and government. This has naturally raised concerns about discrimination encoded in training data, which is subsequently learned by algorithms to perpetuate discriminatory decisions against groups that are protected by law, even in the absence of "discriminatory intent" by those designing and deploying the algorithm. A typical example is an algorithm serving potentially predatory ads to protected groups. Such issues resulted in a 2014 report from the US Executive Office [1] which voiced concerns about discrimination in machine learning and called for additional research to design fair algorithms. The primary question we study in this paper is

How can we maintain high accuracy of a learning algorithm while reducing discriminatory biases?

The measure of fairness we consider is *statistical parity*, which is achieved by definition if the protected subgroup is as likely as the unprotected population to have a given label. As one would expect, if the training data provided to a learning algorithm encodes bias then any success in removing that bias incurs some cost in the classifier's accuracy. Hence, a primary concern is to study the trade-off between bias and accuracy and design algorithms that make favorable trade-offs.

An approach to optimizing this tradeoff is to identify examples for which the learning algorithm is "unsure" of the correct classification, and manipulate the labels of these points so as to improve fairness. Indeed, by changing the labels of "unsure" examples, we can reduce bias and still maintain low error rate, and we show this is the case for three famous learning algorithms: AdaBoost, support vector machines, and logistic regression. Each algorithm provides a measure of confidence in its prediction which we leverage to improve the fairness of the final classification. While we compare a few potential methods for reducing bias, our most successful method uses the confidence values to "shift the decision boundary" for members of the protected class. This technique achieves or outperforms the state of the art on the datasets we test. These datasets provide both a natural interpretation of the method and a quantification of the error-bias tradeoff.

A major challenge in the study of fairness in learning is to find an appropriate definition of what it means for an algorithm to be fair. Presently there is no single accepted definition of fairness. Dwork et al. [2] point out that statistical parity is only a measure of population-wide fairness. In particular, they provide a laundry list of ways one could achieve statistical parity while still exhibiting serious and unlawful discrimination. In addition to analyzing the statistical parity of our methods, we introduce a new notion of fairness that we call *random bias individual fairness* (RBIF). Intuitively, this measure assumes the bias is generated i.i.d. at random, and measures the ability for the learning algorithm to recover the true labels when given the biased training data. We produce unbiased data by generating a new random feature for a given dataset, and then we introduce synthetic bias against that feature.

Although there are several papers on "fair" versions of learning algorithms such as naive Bayes and decision trees, some of the most successful and widely used machine learning algorithms have not been studied in the context of fair learning before. In this paper we consider three ubiquitous learning algorithms: boosting, support vector machines, and logistic regression. Fairness properties of logistic regression have been studied previously by Kamashima et. al [3]; to the best of our knowledge we are the first to study boosting and SVM in this context, and our confidence-based analysis is new for all three algorithms.

The paper is organized as follows. In Section I we review the previous work on fairness and the three algorithms we study. In Section II we define four methods for fair learning as well as *random bias individual fairness*. In Section III we describe our experiments and their results.[1] We close with a discussion in Section IV.

[1]The entire Python source used to generate the diagrams and tables is available at https://www.dropbox.com/sh/p7s6nghntvdxxeo/AABJQIhaH_GDIZJnYAurgufpa?dl=0. This will be updated to a (non-anonymized) Github repository for the final version.

I. Background

A. Notions of fairness

The study of fairness in machine learning is young, but there has been a lot of disparate work studying notions of what it means for data to be fair. See the extensive survey of [4] for a detailed discussion. Still, there is no established measure of *fairness* for a learning algorithm. Two prominent definitions of fairness that have been recently studied in the literature are *statistical parity* and *k-nearest-neighbor consistency.*

Statistical parity: Let D be a distribution over a set of labeled examples X with label $l : X \to \{-1, 1\}$ and a protected subset $S \subset X$. The *bias* of D is defined as the difference in probability of an example in S having label 1 and the probability of an example in S^C having label 1, i.e.

$$B(D, S) = \Pr_{x \sim D|_{S^C}}[l(x) = 1] - \Pr_{x \sim D|_S}[l(x) = 1].$$

The bias of a hypothesis h is the same quantity with $h(x)$ replacing $l(x)$. If a hypothesis has low bias in absolute value we say it achieves *statistical parity.* Note that S represents the group we wish to protect from discrimination, and the bias represents the degree to which they have been discriminated against. The sign of bias indicates whether S or S^C is discriminated against. We use statistical parity for our population-wide fairness measure.

kNN-consistency: Dwork et al. [2] point out that while bias is undesirable, it does not account for all possible forms of discrimination. Rather, it is a measure of group fairness rather than individual fairness. The second notion, due to [2], calls a classifier "individually fair" if it classifies individuals who are close to each other similarly. They use k-nearest-neighbor to measure consistency of close individuals. Note "closeness" is defined with respect to a metric space chosen as part of the data cleaning and feature selection process.

We define a new notion of fairness that departs from previous literature in that it does not require a metric on the underlying space. Rather, it makes the assumption that the process generating the bias is i.i.d. random, and measures the ability for an algorithm to recover the true labels from the biased dataset. We posit that any algorithm which is considered "fair" should recover from i.i.d. random bias against a protected class.

Other fairness measures: Friedler et al. [5] define a measure called *disparate impact* based on the "80% rule" used by some US government institutions and present approaches to reduce it. For more information on different notions of discrimination and fairness of data, we refer the reader to the survey of Romei and Ruggieri [4].

B. Related work on fair algorithms

Learning algorithms studied previously in the context of fairness include naive Bayes [6], decision trees [7], and logistic regression [8]. The two main approaches in most of these papers are massaging and regularization. Massaging means changing the biased dataset before training to remove the bias in the hope that the learning algorithm trained on the now unbiased data will be fair. Massaging is done in the previous literature based on a ranking learned from the biased data. In this paper we do massaging randomly for the sake of simplicity. The regularization approach consists of adding a regularizer to the optimization problem which penalizes the classifier for discrimination.

Two other notable papers in the fairness literature are "Fairness through awareness" ([2]) and "Learning Fair Representations" ([9]). In the former paper the authors describe a framework for maximizing the utility of a classification with the constraint that similar people be treated similarly. One shortcoming of this approach is that it relies on a hypothetical metric on the data that tells us the similarity of individuals with respect to the classification task. As the authors themselves admit, it is unclear how such a metric can be obtained. The authors of the latter paper formulate the problem of fairness in terms of intermediate representations: the goal is to find a representation of the data which preserves as much of the relevant attributes of the original data points as possible while simultaneously obfuscating membership in the protected class.

When applicable, we will include the results of the earlier papers for comparison.

C. AdaBoost

Boosting algorithms work by combining *base hypotheses*, "rules of thumb" that have a fixed edge over random guessing, into highly accurate predictors. In each round, a boosting algorithm will change the weights of the data points and find the base hypothesis that achieves the smallest weighted error on the sample. It always increases the weights of the incorrectly classified examples, thus forcing the base learner to improve the classification of the examples that are the hardest to classify correctly. In this paper we study AdaBoost, a ubiquitous boosting algorithm. The algorithm is given in Algorithm 1. In all of our experiments we boost decision stumps for $T = 20$ rounds.

Algorithm 1 AdaBoost [10]

for $i = 1$ **to** m **do**
  $D_1(i) = \frac{1}{m}$
end for
for $t = 1$ **to** T **do**
  $h_t =$ base hypothesis with smallest error
  $\epsilon_t = \sum_{i=1}^{m} D_t(i)(1 - \delta_{h_t(\mathbf{x}_i), y_i})$
  $\alpha_t = \frac{1}{2} \log \frac{1-\epsilon_t}{\epsilon_t}$
  $Z_t = 2\sqrt{\epsilon_t(1-\epsilon_t)}$
  for $i = 1$ **to** m **do**
    $D_{t+1}(i) = \frac{D_t(i)e^{-h_t(\mathbf{x}_i)y_i}}{Z_t}$
  end for
end for
$g = \sum_{t=1}^{T} \alpha_t h_t$
$h = \text{sgn} \circ g$
return h

Given hypotheses h_i with weights α_i computed by AdaBoost, the *margin* of a labeled data point $(\mathbf{x}, y)$ is

$$\text{margin}(\mathbf{x}, y) = y \frac{\sum_{i=1}^{T} \alpha_i h_i(\mathbf{x})}{\sum_{i=1}^{T} \alpha_i}$$

where α_i is the weight of h_i in the linear combination defined by AdaBoost. We define the similar *signed confidence* of AdaBoost for an unlabeled point x,

$$\mathrm{conf}(\mathbf{x}) = \frac{\sum_{i=1}^{T} \alpha_i h_i(\mathbf{x})}{\sum_{i=1}^{T} \alpha_i}.$$

The absolute values of the two quantities are equal and measure the confidence of AdaBoost in its classification for that particular example. The difference between the two is that whereas the sign of the margin indicates whether the classification is correct, the sign of the confidence tells us the classification itself. Also, the signed confidence can be computed without access to the correct label.

It is well known that the training error of AdaBoost decreases exponentially in the number of rounds, and Schapire et al. [11] prove that the generalization error of AdaBoost can be bounded in terms of the empirical probability of observing a small value of $\mathrm{margin}(\mathbf{x})$ on the training set. This suggests that examples with small confidence are more likely to be incorrect than examples with large margins. In particular, one might hope that one could take advantage of this for fairness by flipping negative labels of members of the protected class with a small confidence. Indeed, is the strategy we analyze in the rest of this paper.

D. Support vector machines

The support vector machines is a framework for learning linear predictors in high (possibly even infinite) dimensional feature spaces. A hyperplane is defined by its normal vector $\mathbf{w}$. If positive and negative examples are separable by a hyperplane, the hyperplane with the largest margin, i.e. the largest distance from the nearest point, is returned. Often the rather strong assumption of linear separability does not hold. In this case we solve a regularized loss minimization problem introduced by Cortes and Vapnik [12] and commonly called Soft-SVM:

$$\min_{\mathbf{w},\xi,b} \left(\frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m} \xi_i \right)$$
$$\text{s.t. } \forall i : y_i(\mathbf{w}\cdot\mathbf{x}_i + b) \geq 1 - \xi \text{ and } \xi_i \geq 0.$$

We will also use the kernel trick, introduced by Boser et al. [13], of implicitly embedding the input space into some higher dimensional feature space. The embedding is defined by a symmetric positive semidifinte function $K(\mathbf{x},\mathbf{x}')$ corresponding to the inner product of the space. In this paper we will use the Gaussian (RBF) kernel $K_{RBF}(\mathbf{x},\mathbf{x}') = \exp\left(-\frac{\|\mathbf{x}-\mathbf{x}'\|^2}{2\sigma^2}\right)$ with parameter $\sigma = 0.1$ for the Census Income and Singles datasets and the linear kernel $K_{lin}(\mathbf{x},\mathbf{x}') = \langle \mathbf{x},\mathbf{x}'\rangle$ for the German dataset (the datasets are described in Section III). The kernel function defines an embedding ψ from the input space into an infinite-dimensional Hilbert space. Let $\mathbf{w}$ denote the normal vector of the hyperplane found by the kernelized SVM in this space. Then we define the *confidence* of SVM as follows:

$$\mathrm{conf}(\mathbf{x}) = \langle \mathbf{w}, \psi(\mathbf{x})\rangle.$$

As in the case of boosting, the confidence has the same magnitude as the analogous SVM margin, but the sign indicates the classification instead of its correctness.

E. Logistic regression

Logistic regression is also a method for learning linear predictors. The classifier output by logistic regression is of the form

$$h(\mathbf{x}) = \mathrm{sign}(\phi(\langle \mathbf{w},\mathbf{x}\rangle) - 1/2)$$

where $\phi(z) = \frac{1}{1+e^{-z}}$ is the logistic function. The vetor $\mathbf{w}$ is found by empirical risk minimization. The loss function used in logistic regression is the logistic loss

$$\ell(\mathbf{w},(\mathbf{x},y)) = \log(1 + e^{-y\langle \mathbf{w},\mathbf{x}\rangle}).$$

The ERM problem associated with regularized logistic regression is

$$\mathrm{argmin}_{\mathbf{w}} \|\mathbf{w}\|^2 + C\sum_{i=1}^{m} \log(1 + e^{-y_i\langle \mathbf{w},\mathbf{x}_i\rangle}).$$

Here we define the confidence of logistic regression simply as the value that the classifier takes before rounding:

$$\mathrm{conf}(\mathbf{x}) = \phi(\langle \mathbf{w},\mathbf{x}\rangle).$$

II. Methods

We define our methods. In what follows X is a labeled dataset, $l(x)$ are the ground truth labels, and $S \subset X$ is the protected group. We further assume that members of S are less likely than S^C to have label 1. First we describe three relabeling algorithms. A relabeling algorithm, when given a hypothesis h and a labeled data set (X,l), produces a new hypothesis h' that flips the output of h according to some rule.

The *random relabeling* (RR) algorithm computes the probability p for which, if members of S with label -1 under h are flipped by h' to $+1$ randomly and independently with probability p, the bias of h' is zero in expectation. The classifier h' is then defined as the randomized classifier that flips members of S with label -1 with probability p and otherwise is the same as h.

Next, we define *random massaging* (RM). Massaging strategies, introduced by [14], involve eliminating the bias of the training data by modifying the labels of data points, and then training a classifier on this data in the hope that the statistical parity of the training data will generalize to the test set as well. In our experiment, we massage the data randomly; i.e. we flip the labels of S from -1 to $+1$ independently at random with the probability needed to achieve statistical parity in expectation, as in RR.

Our main contribuition is the following algorithm which we will call *shifted decision boundary* (SDB). This is a relabling algorithm which takes confidence into consideration. The classification of a data point by any of the three learning algorithms studied in this paper is a function of the confidence:

the predicted label is positive if and only if the confidence is above a fixed threshold. This threshold is 0 for AdaBoost and SVM and $\frac{1}{2}$ for logistic regression. Also, for these algorithms, the classification is more likely to be correct for data points with high absolute confidence. Consequently by changing the labels of data points in the protected group which have small negative confidence, not only do we reduce bias, but we likely do not increase label error significantly since these points are likely to be misclassified by h. Relabeling points with small negative confidence is equivalent to shifting the confidence threshold down: indeed, this is what our algorithm will do for points in the protected group.

More formally, the SDB algorithm computes the value θ such that bias is minimized by shifting the minimum required signed confidence for examples from S from zero to θ. That is, SDB returns a classifier h' of the following form: $x \in S$ then $h'(x) = 1$ iff $\text{conf}(x) \geq \theta$, and otherwise $h'(x) = h(x)$. Then θ is chosen to minimize the bias of h' on S.

Finally, in *fair weak learning* (FWL) we replace a standard boosting weak learner with one which tries to minimize a linear combination of error and bias and run the resulting boosting algorithm unchanged. The weak learner we use computes the decision stump which minimizes the sum of label error and bias of its induced hypothesis. Fair weak learning is only applicable to boosting.

Now we define our new method for evaluating the fairness of an algorithm. As we noted in Section I, previous notions of fairness all suffer from one of the following two problems: either they only measure fairness in statistical terms over the entire group or, if they aim to measure individual fairness, they rely on additional information about the data (such as a metric) which is usually not available explicitly.[2]

To construct a fairness measure that does not suffer from these limitations, we introduce synthetic bias to the data. The advantage of synthetically generated bias is that in this case we know the original, unbiased ground truth, and therefore we can measure the performance of the learning algorithm against this ground truth. In addition, we guarantee by design that we avoid the adversarial targeted discrimination types listed in [2].

In particular, we test how these algorithms are resistant to random noise that introduces bias against a random subset of the individuals. This is formalized as follows:

Definition 1. *We define the* random bias individual fairness *(RBIF) of a learning algorithm A on a labeled dataset X, l as follows. Introduce a new uniformly random binary feature z on elements of X. Flip the labels of examples x that have $z = 0$ independently with probability p to -1 to get a new dataset X', l'. Run A on X', l' and let h be the resulting hypothesis. The random bias individual fairness of A is the expected fraction of flipped examples $x \in X'$ for which $h(x) = l(x)$.*

In our experiments we set $p = 0.2$. RBIF can be thought of as the following experiment: A learning algorithm is given a dataset in which bias has been generated at random. That is, we change the labels of a few individuals based on a feature which is random with respect to the classification task. The goal of the algorithm is to then recover the ground truth labels in the original dataset. This models the probability that an individual who is subject to random bias will be treated fairly.

This definition naturally generalizes to an arbitrary distribution over examples, and we defer the analysis of such a definition to future work.

[2]Moreover, the work in [2] suggests that learning a suitably fair similarity metric from the data is as hard as the original problem.

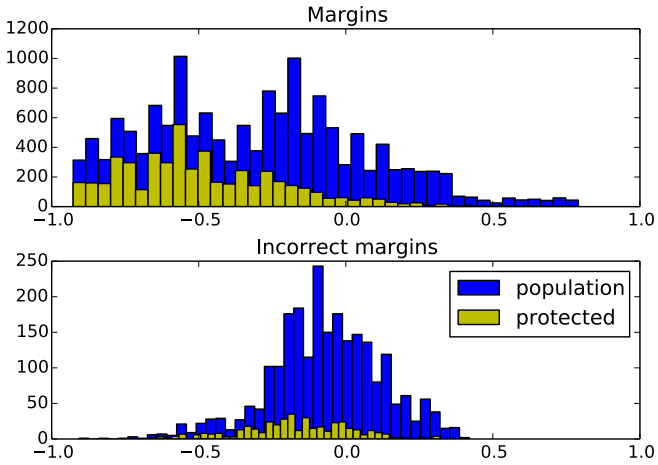


Fig. 1: Histogram of boosting confidences for the Census data set. The vast majority of women are classified as -1, and the incorrect classifications are closer to the decision boundary.

III. Experiments

A. Datasets

The Census Income dataset [15], extracted from the 1994 Census database, contains demographic information about 48842 American adults. The prediction task is to determine whether a person earns over \$50K a year. The dataset contains $16,192$ females (33%) and $32,650$ males. Note 30.38% of men and 10.93% of women reported earnings of more than \$50K, therefore the bias of the dataset is 19.45%. Further note that since 76% of the data points have negative labels, the constant -1 classifier achieves 76% accuracy and perfect statistical parity. This baseline does not appear to have been considered in previous literature for fair algorithms which used this dataset.

The German credit dataset [15] contains financial information about 1000 individuals who are classified into groups of good and bad credit risk. The "good" credit group contains 699 individuals. Following the work of [14], we consider age as the protected attribute with a cut-off at 25. Only 59% of the younger people are considered good credit risk, whereas of the 25 or older group, 72% are creditworthy. This gives us a bias of 13%.

In the Singles dataset, extracted from the marketing dataset of [16], the goal is to predict whether annual income of a household is greater than \$25K from 13 other demographic attributes. The protected attribute is gender. The dataset contains $3,653$ data points, $1,756$ (48%) of which belong to the protected group. 34% of the dataset has a positive label. We extracted the subset of records whose respondents identified as "single."

B. Results and analysis

In this section we state our experimental results. They are summarized in Tables I, II, and III for the Census Income, German, and Singles datasets, respectively. For comparison, we also included the numbers for the Learning Fair Representations (LFR) method of [9] for the Census Income dataset and also the numbers for the Classification with No Discrimination (CND) method of [14] (these numbers were estimated from figures in the paper since they were reported graphically). In the former paper, the authors implemented three other learning algorithms, these are unregularized logistic regression, Fair Naive-Bayes [14], and Regularized Logistic Regression [3]. These methods all had errors above 20%; thus we see that our confidence-based relabeling methods outperform the state of the art for the Census Income dataset. To investigate the trade-offs made by these relabeling methods more closely, Figures 2, 3, and 4 show the rate at which error increases as bias goes to zero.

For the Census Income dataset, AdaBoost and logistic regression with SDB have the best performance. Both SDB algorithms achieve statistical parity with about 18% error. Logistic regression with RM achieves slightly lower error (17.96%) with -2.25% bias. Moreover, the two SDB algorithms have the highest RBIF. To the best of our knowledge, no other published method has achieved statistical parity with less than 20% error.

The different methods have similar performance on the German dataset with accuracy mostly in the 24-27$\%$ range. Variants of logistic regression have the smallest error. As Figure 3 shows, label error practically stays constant as the decision boundary is shifted.

In the case of the Singles dataset, SDB for SVM is the clear winner. It is notable that not only is there no significant increase in the error compared to the unmodified SVM baseline, but error actually decreases slightly as the confidence threshold is shifted for the protected group. (This can also be seen in Figure 4c.)

Note again the difference in SVM kernels between the datasets. The Gaussian kernel performs well for the Census Income and Singles dataset. However, in the case of the German dataset, which is the smallest of the three, with the Gaussian kernel every point becomes a support vector. This is not only a clear sign of overfitting but also makes SDB impossible since the model gives the same confidence for almost every data point.

We found that fair weak learning (FWL) does empirically reduce bias, but does not achieve statistical parity. Moreover, the label error of FWL is not better than that of SDB, and the trade-off between label error and bias cannot easily be controlled. The same is true for random massaging. On the other hand, we can see from Table II that at least in some cases RM and FWL perform equally well with SDB, though in most cases one or both methods are far worse.

A further advantage of SDB is that the trade-off between label error and bias can be controlled after training. To decide how much bias and error we want to allow, we do not have to pick a hyperparameter before training the algorithm, unlike for most other fair learning methods. This means that the computational cost of choosing the best point on the trade-off curve is very low, and the tradeoff is transparent. These results show the advantages of SDB: the confidence can be used to find a superior classifier.

The results also show the usefulness of RBIF as a measure of fairness. As we can see in Tables I and III, even when random relabeling slightly outperforms SDB in terms of label error, SDB often beats RR by as much as ten percent RBIF. This suggests that the performance of fair learning algorithms should not be evaluated solely by their accuracy and bias.

A natural baseline for RBIF is 0.5, since a hypothesis chosen uniformly at random will flip back half of the points that were flipped to -1. We see that the RBIF of the unmodified learning algorithms, with the exception of SVM and AdaBoost on the German dataset, is almost always below 0.5, showing that the classifiers encode the bias introduced into the training data. Our evidence that we can recover from randomly introduced bias while still achieving low label error is promising. Even though the RBIF of the fair learning algorithms is often still below 0.5, they increase RBIF significantly compared to the unmodified baseline, sometimes almost doubling it.

IV. Discussion

In this paper, we introduce a general method for balancing discrimination and label error. This method, which we call shifted decision boundary (SDB), is applicable to any learning algorithm which has an efficiently computable measure of confidence. We study three such algorithms, AdaBoost, SVM, and linear regression, compare our methods to other methods proposed in the earlier literature and our own baselines, and empirically evaluate our methods' performance on three datasets. We find that SDB generally outperformed our other methods and the state of the art, and we provide theoretical justification for its success.

We define the RBIF measure, and our empirical results suggest its usefulness in measuring an algorithm's fairness. Although i.i.d. random bias is a simplified and admittedly unrealistic model of real-world discrimination, we posit that any algorithm which can be considered fair must be fair with respect to RBIF. Moreover, RBIF generalizes to an arbitrary distribution over the input data. We leave the theoretical analysis of this generalization to future work.

Finally, we have not analyzed measures such as SDB and RBIF from a legal or sociological perspective. Despite achieving low bias and high accuracy, SDB generalizes (and can be interpreted as) "lowering the bar" for the protected class. Such methodologies are controversial in practice, and it is not clear to what extent shifting a decision boundary of a learning algorithm, which may be composed of a complex combination of features, constitutes "lowering the bar." Likewise, it would be interesting to relate RBIF to legal notions of fairness.

	SVM	SVM (RR)	SVM (SDB)	SVM (RM)	LFR [9]
label error	0.1471 (3.2e-33)	0.2006 (2.9e-06)	0.2198 (0.0)	0.1749 (7.1e-06)	0.2299
bias	0.1689 (3.2e-33)	-0.0179 (2.1e-05)	-0.1550 (8.5e-34)	0.0713 (8.2e-05)	0.0020
RBIF	0.2702 (2.0e-04)	0.2821 (2.0e-04)	0.2925 (2.0e-04)	0.2706 (2.2e-04)	n/a
	LR	LR (RR)	LR (SDB)	LR (RM)	
label error	0.1478 (2.3e-07)	0.2056 (4.7e-06)	0.1828 (4.1e-6)	0.1796 (1.8e-05)	
bias	0.1968 (1.1e-05)	-0.0046 (3.3e-05)	-0.0068 (2.0e-5)	-0.0225 (3.0e-04)	
RBIF	0.4647 (1.7e-04)	0.4637 (5.9e-04)	0.5554 (4.9e-04)	0.4361 (2.0e-04)	
	AdaBoost	AB (RR)	AB (SDB)	AB (RM)	AB (FWL)
label error	0.1529 (4.8e-06)	0.2073 (1.2e-05)	0.1828 (1.4e-05)	0.1888 (5.3e-05)	0.1820 (3.6e-05)
bias	0.1856 (1.4e-04)	-0.0025 (3.8e-05)	-0.0036 (1.4e-05)	-0.0283 (1.5e-03)	0.0691 (0.0001)
RBIF	0.4372 (1.0e-03)	0.4645 (3.2e-04)	0.5340 (9.4e-04)	0.421 (6.8e-04)	0.5174 (0.0004)

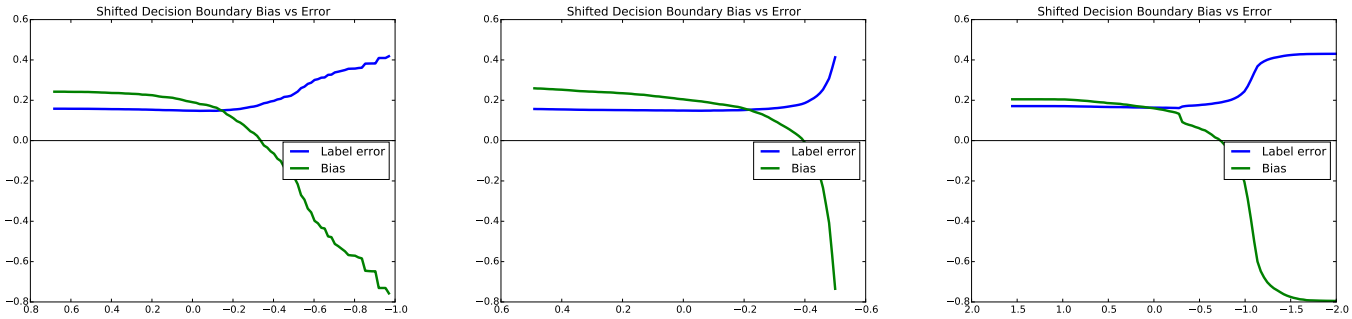
TABLE I: A summary of our experimental results for the Census Income data for relabeling, massaging, and the fair weak learner. The threshold for SDB was chosen to achieve perfect statistical parity on the training data. The variances are in parentheses.

	SVM	SVM (RR)	SVM (SDB)	SVM (RM)	CND [14]
label error	0.2823 (0)	0.2802 (2.2e-05)	0.2742 (0.00073)	0.2784 (0.00015)	0.2757 (0.026)
bias	0.0886 (1.8e-33)	-0.0302 (0.0015)	-0.0530 (0.0014)	-0.0821 (0.00099)	0.0327 (0.0020)
RBIF	0.6756 (0.0065)	0.8218 (0.0059)	0.8011 (0.0095)	0.6401 (0.0097)	n/a
	LR	LR (RR)	LR (SDB)	LR (RM)	
label error	0.2541 (2.1e-05)	0.2489 (3.7e-05)	0.2538 (0.00016)	0.2495 (8.5e-05)	
bias	0.1383 (0.0002)	-0.0605 (0.0020)	-0.0942 (0.0016)	-0.1006 (0.00066)	
RBIF	0.3070 (0.0045)	0.8791 (0.0024)	0.8659 (0.0032)	0.6599 (0.0034)	
	AdaBoost	AB (RR)	AB (SDB)	AB (RM)	AB (FWL)
label error	0.2602 (8.3e-05)	0.2589 (0.00019)	0.2622 (0.00032)	0.2576 (0.00016)	0.2520 (0.0001)
bias	0.2617 (0.0074)	-0.0310 (0.0022)	-0.0539 (0.0025)	0.0211 (0.0042)	0.2355 (0.0084)
RBIF	0.6774 (0.0048)	0.8854 (0.0048)	0.8406 (0.0065)	0.6633 (0.0040)	0.6935 (0.0069)

TABLE II: A summary of our experimental results for the German data for relabeling, massaging, and the fair weak learner. The threshold for SDB was chosen to achieve perfect statistical parity on the training data. The variances are in parentheses.

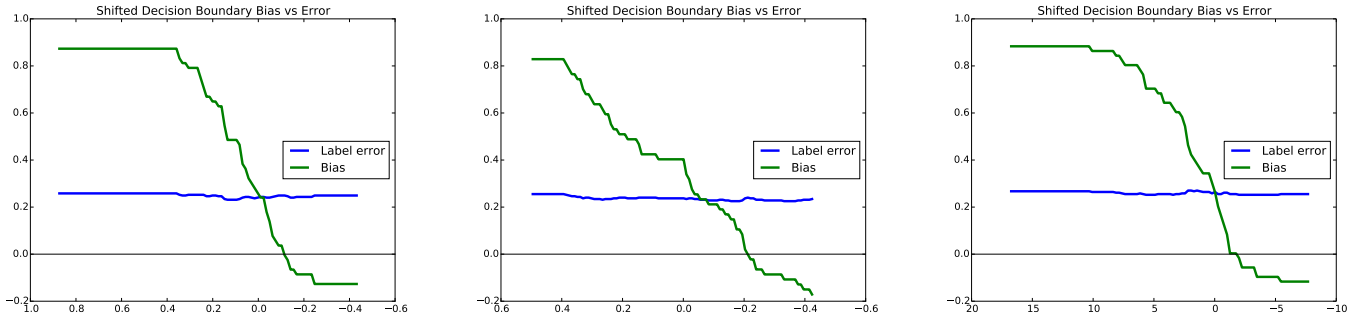
	SVM	SVM (RR)	SVM (SDB)	SVM (RM)	
label error	0.2718 (3.2e-33)	0.2919 (1.5e-05)	0.2672 (5.3e-05)	0.2750 (1.7e-04)	
bias	0.0550 (2.0e-34)	0.0311 (1.3e-04)	0.0039 (4.4e-04)	0.0209 (8.4e-04)	
RBIF	0.2424 (2.0e-03)	0.2931 (1.8e-03)	0.3016 (9.7e-04)	0.2175 (2.2e-03)	
	LR	LR (RR)	LR (SDB)	LR (RM)	
label error	0.2742 (1.3e-32)	0.3165 (3.6e-05)	0.2850 (7.2e-05)	0.2824 (7.3e-06)	
bias	0.1468 (1.0e-34)	0.0054 (2.5e-04)	-0.0090 (2.7 e-04)	0.0556 (9.6e-05)	
RBIF	0.1971 (1.3e-03)	0.2603 (1.3e-03)	0.3606 (4.6e-03)	0.1905 (3.0e-03)	
	AdaBoost	AB (RR)	AB (SDB)	AB (RM)	AB (FWL)
label error	0.2690 (1.4e-05)	0.2914 (1.1e-04)	0.2762 (6.5e-05)	0.2713 (5.3e-06)	0.2705 (4.8e-05)
bias	0.0966 (4.1e-04)	0.0254 (5.6e-05)	0.0100 (7.8e-04)	0.0130 (1.1e-04)	-0.0259 (1.1e-04)
RBIF	0.2864 (3.2e-03)	0.3337 (3.2e-03)	0.4353 (2.4e-03)	0.2807 (2.0e-03)	0.2606 (3.4e-03)

TABLE III: A summary of our experimental results for the Singles data for relabeling, massaging, and the fair weak learner. The threshold for SDB was chosen to achieve perfect statistical parity on the training data. The variances are in parentheses.



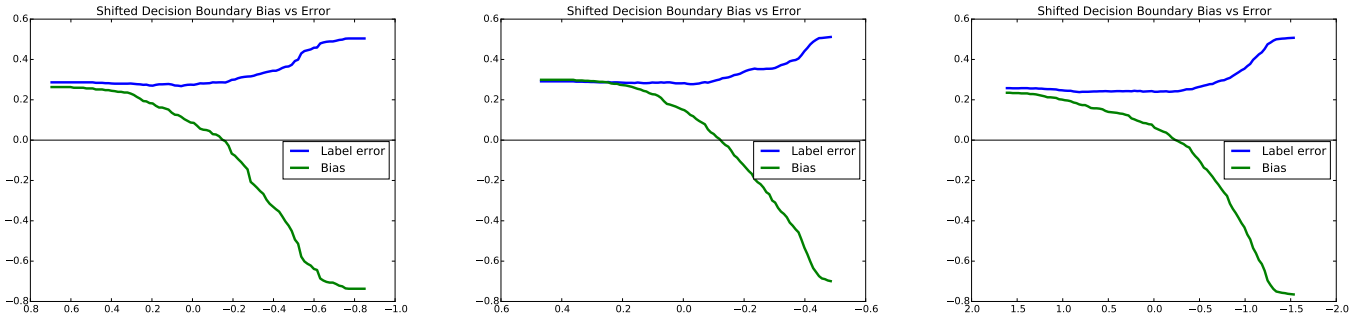
(a) Boosting (b) Logistic Regression (c) SVM

Fig. 2: Trade-off between (signed) bias and error for SDB on the Census Income data. The horizontal axis is the threshold used for SDB.



(a) Boosting (b) Logistic Regression (c) SVM

Fig. 3: Trade-off between (signed) bias and error for SDB on the German data. The horizontal axis is the threshold used for SDB.



(a) Boosting (b) Logistic Regression (c) SVM

Fig. 4: Trade-off between (signed) bias and error for SDB on the Singles data. The horizontal axis is the threshold used for SDB.

REFERENCES

[1] J. Podesta, P. Pritzker, E. J. Moniz, J. Holdren, and J. Zients, "Big data: Seizing opportunities, preserving values," 2014. [Online]. Available: https://www.whitehouse.gov/sites/default/files/docs/big_data_privacy_report_may_1_2014.pdf

[2] C. Dwork, M. Hardt, T. Pitassi, O. Reingold, and R. Zemel, "Fairness through awareness," in *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*. ACM, 2012, pp. 214–226.

[3] T. Kamishima, S. Akaho, and J. Sakuma, "Fairness-aware learning through regularization approach," in *Data Mining Workshops (ICDMW), 2011 IEEE 11th International Conference on*. IEEE, 2011, pp. 643–650.

[4] A. Romei and S. Ruggieri, "A multidisciplinary survey on discrimination analysis," *The Knowledge Engineering Review*, vol. 29, pp. 582–638, 11 2014.

[5] S. Friedler, C. Scheidegger, and S. Venkatasubramanian, "Certifying and removing disparate impact," *arXiv preprint arXiv:1412.3756*, 2014.

[6] T. Calders and S. Verwer, "Three naive bayes approaches for discrimination-free classification," *Data Mining and Knowledge Discovery*, vol. 21, no. 2, pp. 277–292, 2010.

[7] F. Kamiran, T. Calders, and M. Pechenizkiy, "Discrimination aware decision tree learning," in *Data Mining (ICDM), 2010 IEEE 10th International Conference on*. IEEE, 2010, pp. 869–874.

[8] T. Kamishima, S. Akaho, H. Asoh, and J. Sakuma, "Fairness-aware classifier with prejudice remover regularizer," in *Machine Learning and Knowledge Discovery in Databases*. Springer, 2012, pp. 35–50.

[9] R. Zemel, Y. Wu, K. Swersky, T. Pitassi, and C. Dwork, "Learning fair representations," in *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, 2013, pp. 325–333.

[10] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *J. Comput. Syst. Sci.*, vol. 55, no. 1, pp. 119–139, 1997.

[11] R. E. Schapire, Y. Freund, P. Bartlett, and W. S. Lee, "Boosting the margin: A new explanation for the effectiveness of voting methods," *Annals of statistics*, pp. 1651–1686, 1998.

[12] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[13] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proceedings of the fifth annual workshop on Computational learning theory*. ACM, 1992, pp. 144–152.

[14] F. Kamiran and T. Calders, "Classifying without discriminating," in *Computer, Control and Communication, 2009. IC4 2009. 2nd International Conference on*. IEEE, 2009, pp. 1–6.

[15] M. Lichman, "UCI machine learning repository," 2013. [Online]. Available: http://archive.ics.uci.edu/ml

[16] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning*. Springer, 2009, vol. 2, no. 1.